Centro Universitario de Ciencias Exactas e Ingenierías

Propuestas del lenguaje en ensamblador

1. **Ramírez Vázquez Ezequiel Alberto**
2. **De La Cruz Chavarria Christian Israel**
3. **Mariscal Viorato Maria Fernanda**
4. **Oscar Beltran Villegas**

Propuesta 1

Esta primera propuesta del lenguajeen ensablador es un algoritmo un tanto conocido, se llama "el problema de las N-Reinas". Basicamense se trata de colocar una reina en cada fila, normalmente entre 4 u 8 reynas, de tal manera que ninguna se ataque, ya que como bien sabemos, este algoritmo se inspira en el tablero del ajedrez, por lo que en el jeugo la pieza denominada como "Reina" tiene la libertad de moverse a voluntad. Para comenzar con este algoritmo debemos colocar una reina en una casilla de su fila y ¿como se hace esto?. Primero generamos un vector. Donde colocaremos n reinas donde cada posición será donde colocaremos una reina

	0	1	2	3
0				
1				
2				
3				

Imaginando que cada no de los espacios rellenados es una reina esto en otros términos seria que de las 0 hasta la n, representando a la fila donde se pondría la reina en la columna i.

El algoritmo se tratara de un procedimiento recursivo de N-Reinas (int reina) que sera llamada por primera vez como N-Reinas[1] con la idea de que coloque la primera reina y luego se vaya llamando de forma recursiva para colocar la reina 2 y sucesivamente.observamos que cada reina se coloca siempre en la fila i.

En este ejercicio se desarrollan algunos métodos de fuerza bruta (por asi decirlo) y esto se utiliza para resolver el problema de las N-Reinas. Para esto se tienen cuatro condiciones:

a) Utilizar esta técnica de la fuerza bruta de manera aleatoriamente para poder resolver el ejercicio de las n-reinas

b) Utilizar una técnica exhaustiva (esto quiere decir que se tienen que probar todas las combinaciones posibles de las N-Reinas que se pudieran llegar a tener en un tablero de 4 u 8 columnas.

c) En este punto se preguntaran el ¿Por qué el método de la fuerza bruta exhaustivamente? Bueno, este método aunque podría no ser apropiado

para este ejercicio, es el más factible y el que nos puede dar una solución más rápida.

d) debemos Comparar y contrastar el problema de la fuerza bruta aleatoria con el de la fuerza bruta exhaustiva.

Algoritmo recursivo

1. Obviamente, colocaremos una reina en cada fila
2. Se coloca una reina en una casilla de su fila y, a continuación, se intentan colocar las reinas restantes.
3. Si las demás reinas no se pueden colocar con éxito, probamos a colocar la reina actual en otra columna. Caso base: Cuando no quedan reinas por colocar. Además se generan las constantes simbólicas. Se Compruba si una reina está bien colocada. La reina de la fila i está bien colocada si no está en la columna ni en la misma diagonal que cualquiera de las reinas de las filas anteriores. Su programa principal se centra en el siguiente código.:

```
void main (int argc, char *argv[])
{
  int *reinas;  // Vector con las posiciones de las reinas de cada fila
  int nreinas;  // Número de reinas
  int i;        // Contador

  nreinas = -1;

  if (argc==2)
     nreinas = atoi(argv[1]);
  if (nreinas>0) {
      reinas = (int*) malloc ( nreinas*sizeof(int) );
      for (i=0; i<nreinas; i++)
          reinas[i] = -1;
      colocarReina(0,reinas,nreinas);
```

Propuesta 2

Para el desarrollo de este programa, que es un programa de instrucción asistida por computadora se genera para enseñar a un estudiante de escuela primaria las operaciones básicas de matemáticas, los cuales son, sumas restas y multiplicaciones.

El programa nos ayudara y permitirá que el usuario que este usando el código introduzca su nivel de capacidad escolar (como bien se mencionó antes, este código es para enseñar a estudiantes de primaria). El primer nivel de 1 significa que en el programa se debe usar únicamente números de un solo dígito en la realización de algún problema. El nivel 2 significa que en el programa se deben utilizar dos dígitos como máximo, y así sucesivamente con todos los demás niveles.

Este programa se puede usar para permitir al usuario que elija el tipo de problema matemático que el usuario desea estudiar. Ahora bien, volviendo a las opciones, la opción 1 hace referencia a problemas de suma solamente, la opción 2 hace referencia a un problema de resta, la opcion 3 se refiere a los problemas de multiplicación, la opcion 4 se refiere a los problemas de división y por ultimo la opción 5 hace referencia a una mezcla aleatoria de problemas de cualquiera de estos tipos.

Su principal algotirmo es el signietnte:

```
int main()
 {    // abre main
 int s;
 cout<<"Determinacion de numeros perfectos entre 1 y 1000."<<endl;
 for (int counter = 1; 1000 >= counter;++counter)
 {  // Abre for
 s = Perfect (counter);
 if (1 == s)
 PrintPerfect (counter);
 }  // Cierra for
 cout <<endl << endl;
 return 0;
 }   // cierra main
 int Perfect (int n)
 {
 int suma = 0, factor;
 for (int divisor = 1; divisor < n; ++ divisor)
```